Phishing Project

Name : Deriche Yanis

Part I:

A Complete Phishing Simulation with Gophish

What is Phishing?

Phishing is a social engineering attack used to obtain user information such as login credentials and credit card information. It happens when a malicious actor pretends to be someone or something trustworthy to trick a victim into opening an email, IM, or text message.

Scenario

Consider a small business with 30 employees that use Outlook or G-Suite for email, a leading email filtering technology to analyze the content of incoming emails, and a well-known EDR on each endpoint.

Yes, it is possible and interesting to spend some time looking for a way to circumvent a specific email filtering solution, ensuring that your macro-infected Word document avoids modern EDRs and installs persistent malware on the target endpoint.

But first, let us determine how thin the iceberg we are walking on is:

The EDR/email filtering solution may continue to block the payload.

The alerts for “Enable Content” may alarm the end user.

Even if we can successfully install our persistence, it must choose the correct method to “phone home” to the C2 without being blocked by Proxy, Firewall, or raising suspicions.

Word macros accessed in OneDrive via the browser are unlikely to execute.

What is evilginx2?

The evilginx2 framework is a complex Golang Reverse Proxy that allows victims to be proxied against legitimate services while recording credentials and authentication sessions. The sessions collected can then be used to authenticate victim accounts while avoiding 2FA safeguards fully.

Furthermore, in “Cloud-oriented” enterprises like the one described above, a session with a key cloud provider can frequently result in access to sensitive shared folders, organizational data, and even VPN connection details, granting access to the internal corporate network.

evilginx2 employs the reverse proxying concept to efficiently route traffic between phished users (e.g., targeted employees) and legitimate websites (e.g., authentication providers).

It also includes TLS termination capabilities similar to those found in Burp Suite, which means that the user viewing the evilginx2 URL will see green-lock HTTPS browsing while the evilginx2 server decrypts that data and establishes its own new HTTPS connection with the target website.

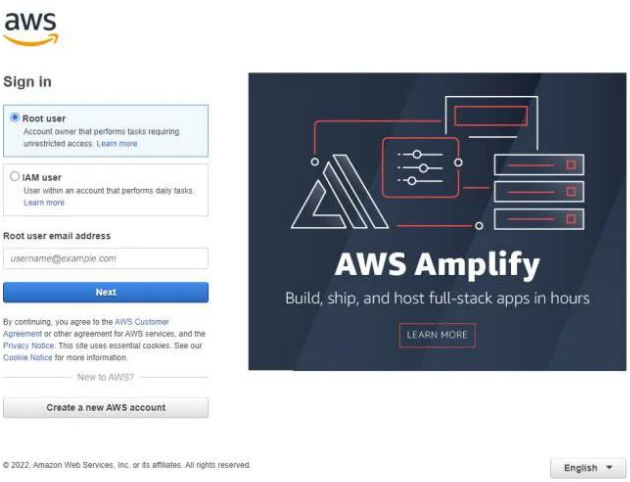
Because of this positioning, evilginx2 can act as a man-in-the-middle and obtain raw credential information without requiring the attacker to clone fake sites or perform heavy lifting. Aside from managing the server, the majority of the configuration is contained in the evilginx2 YAML files, which instruct it on how to behave on a per-domain basis.

Tools to Install

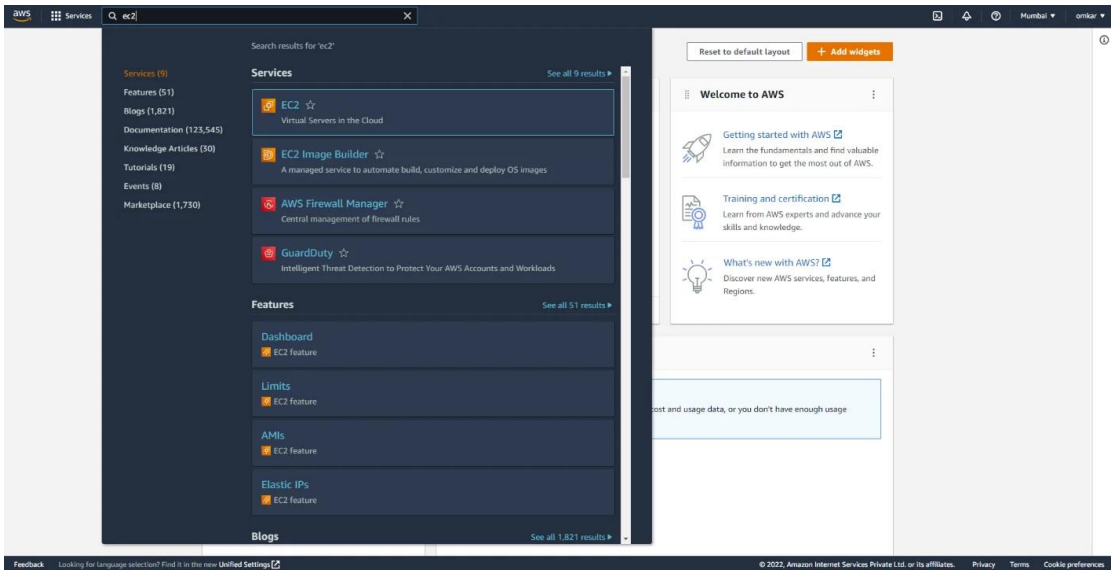
- Route53 (Domain Registration)
- AWS EC2 Instance
- gophish
- evilginx2
- Mailgun

Setting up Amazon EC2

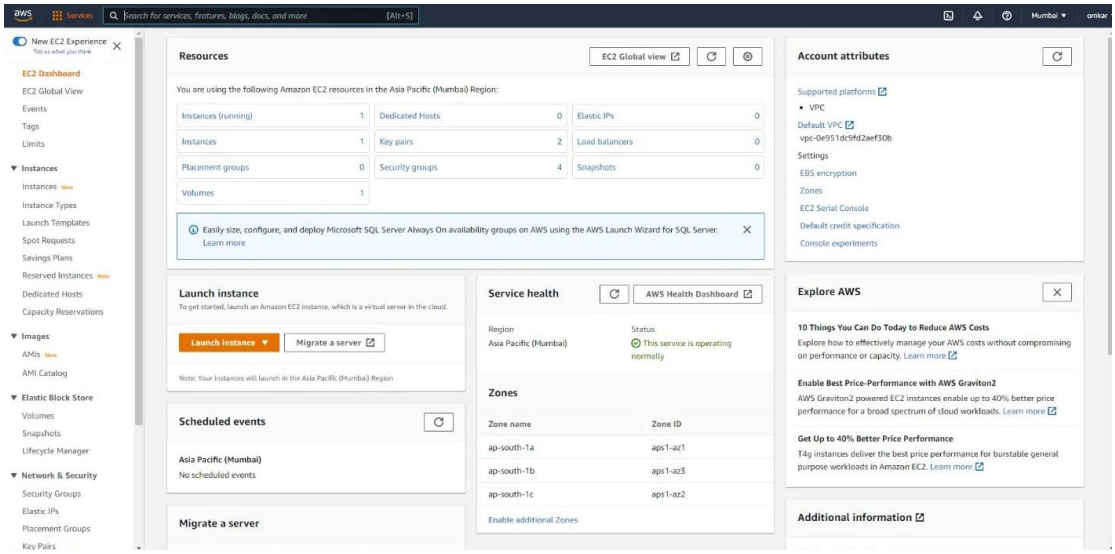
First, Login into AWS Console.



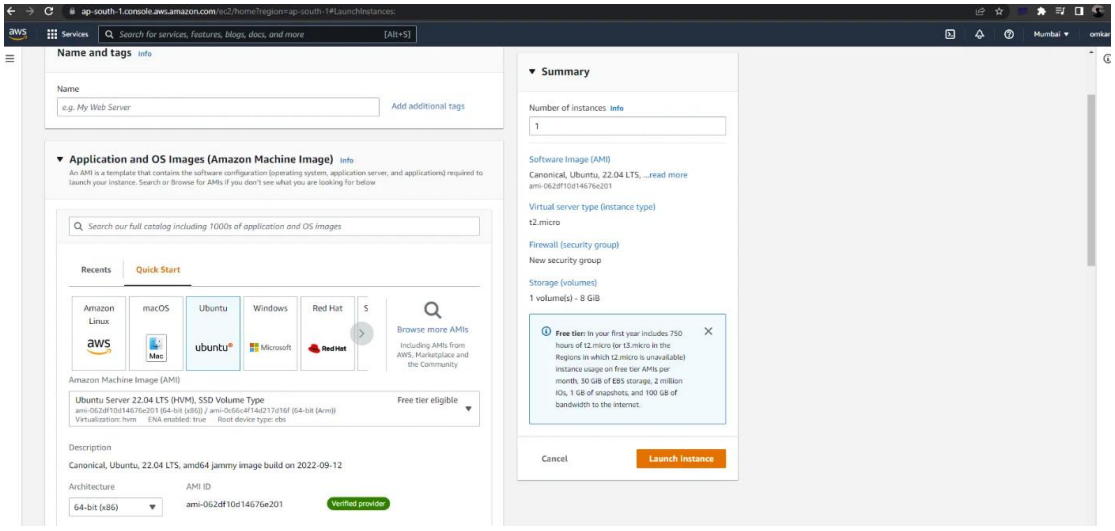
Next, search for the EC2 service.



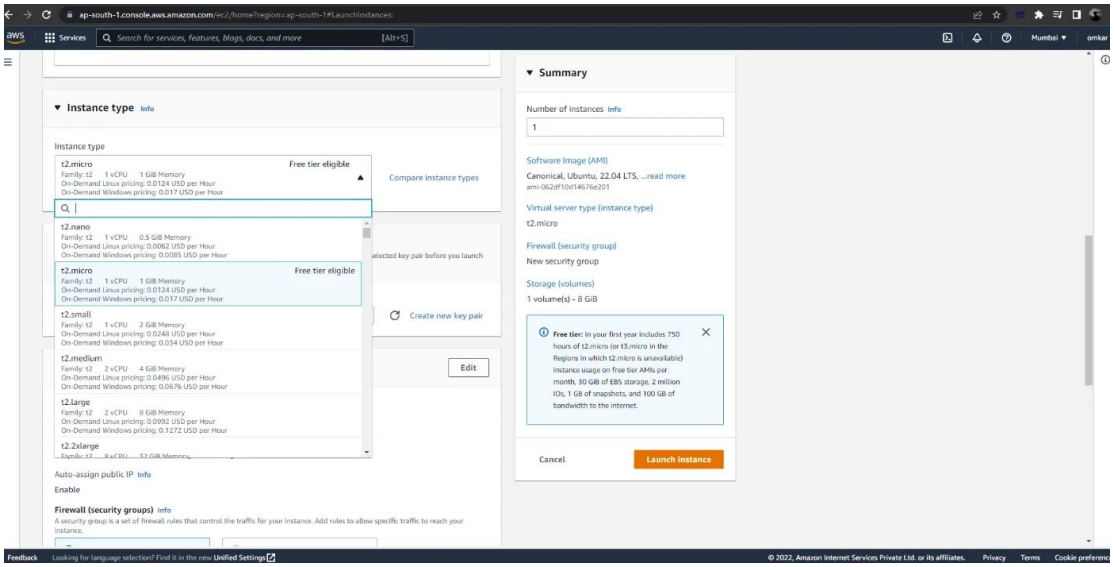
Click on “Launch Instance”.



Select Ubuntu Server (Free tier eligible) and continue as shown below.

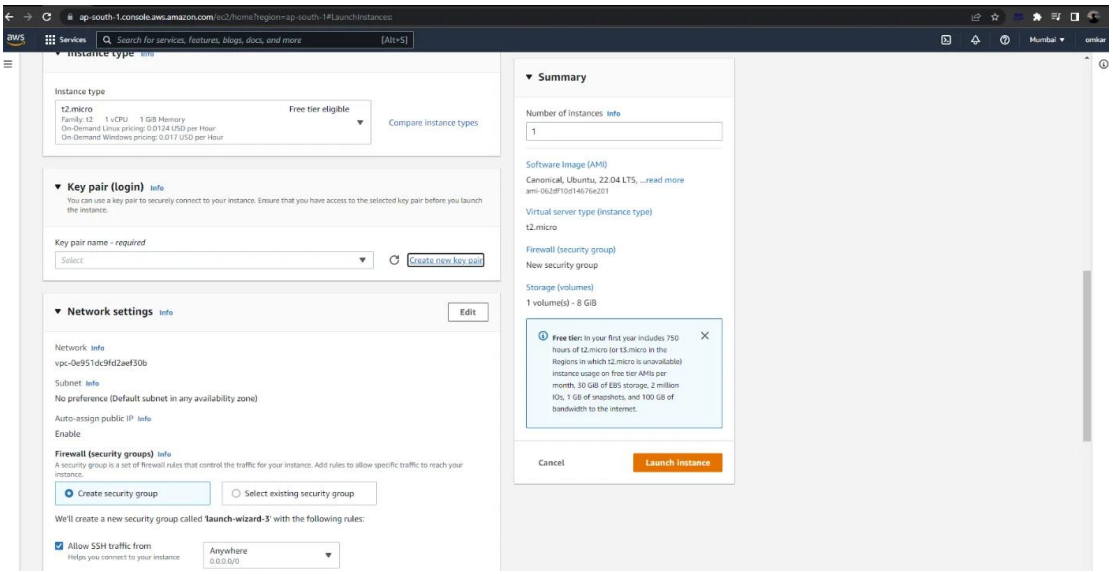


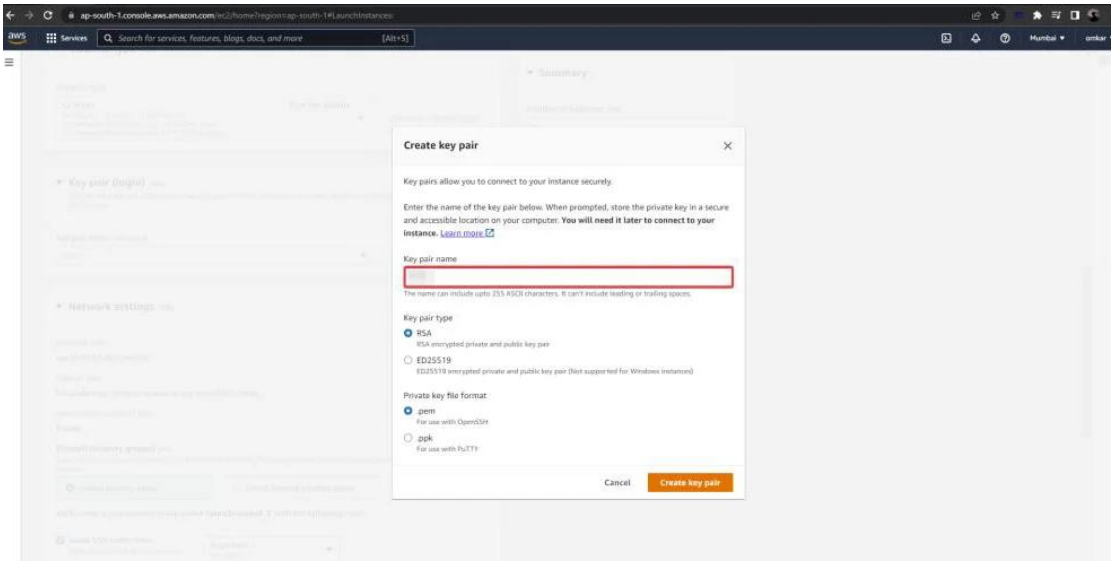
Next, for “Instance type” select “t2.micro”, which is available for free.



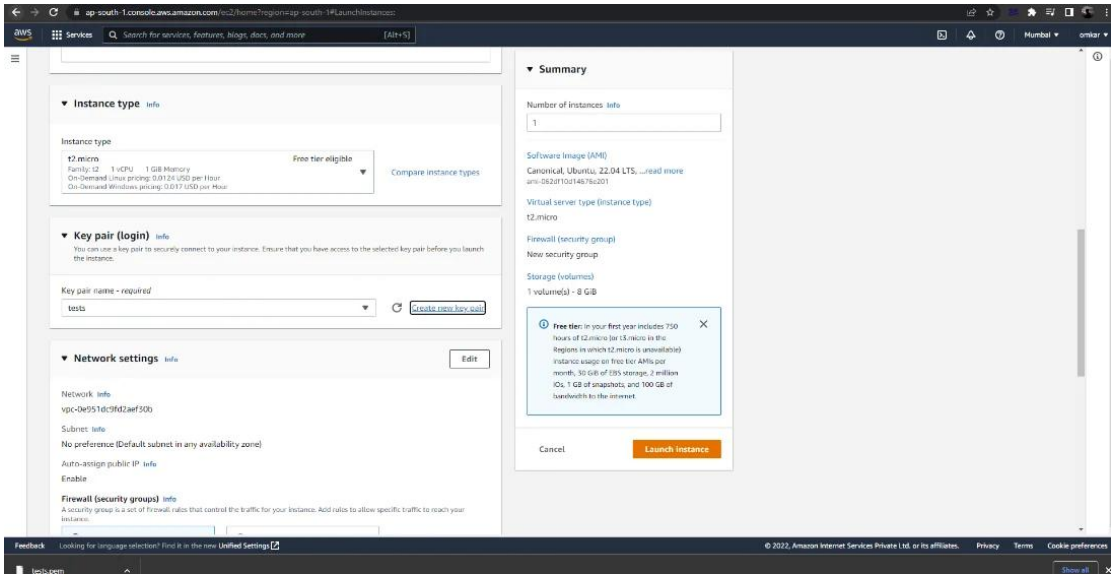
Note: It prompts the user to generate or use an existing private key for SSH access to the Ubuntu system.

Click on “Create a new key pair” and give it a suitable name.

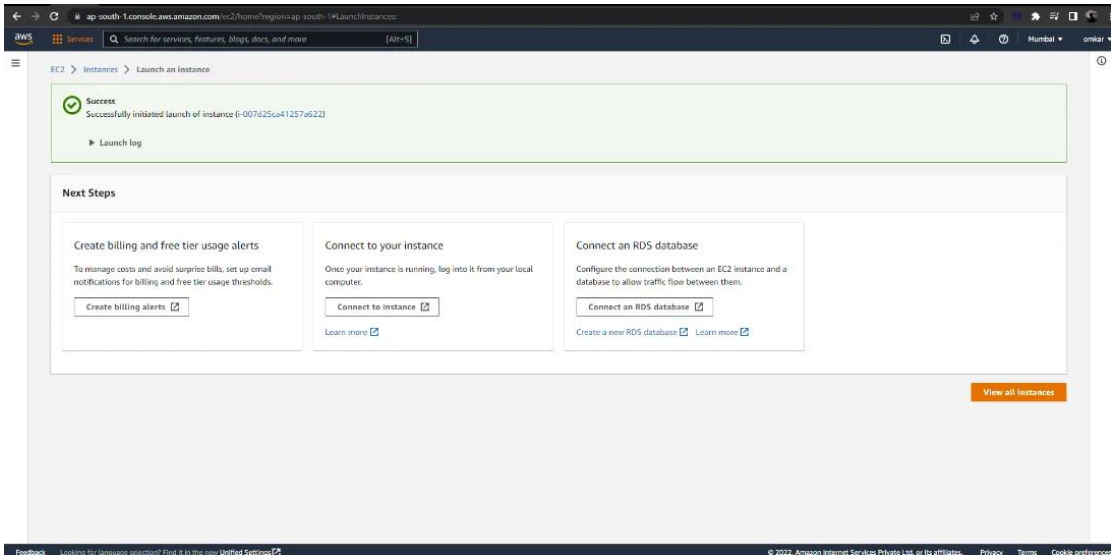




Note that the key pair will be downloaded automatically.

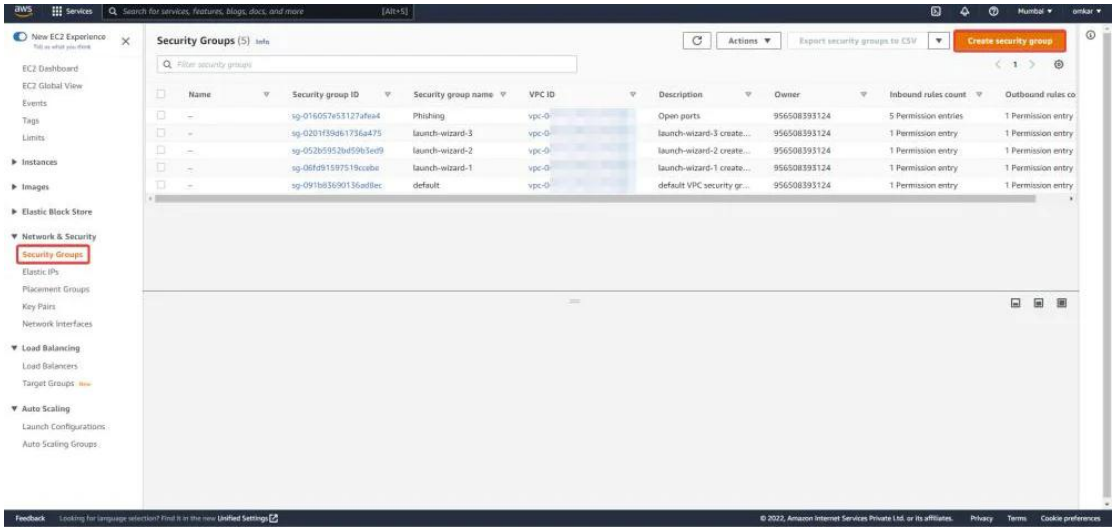


Launch the instance after downloading the private key file.

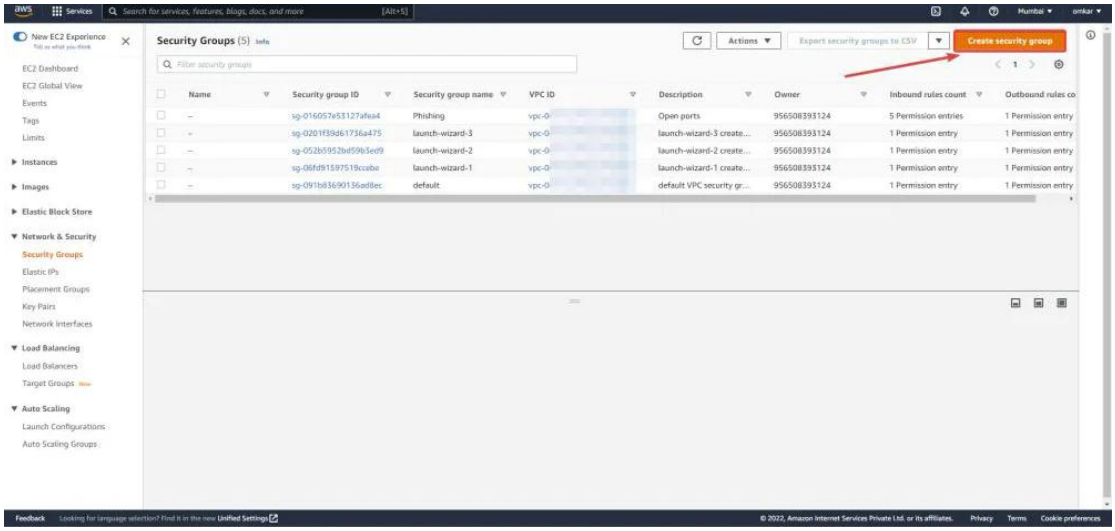


Configuring a Custom Security Group

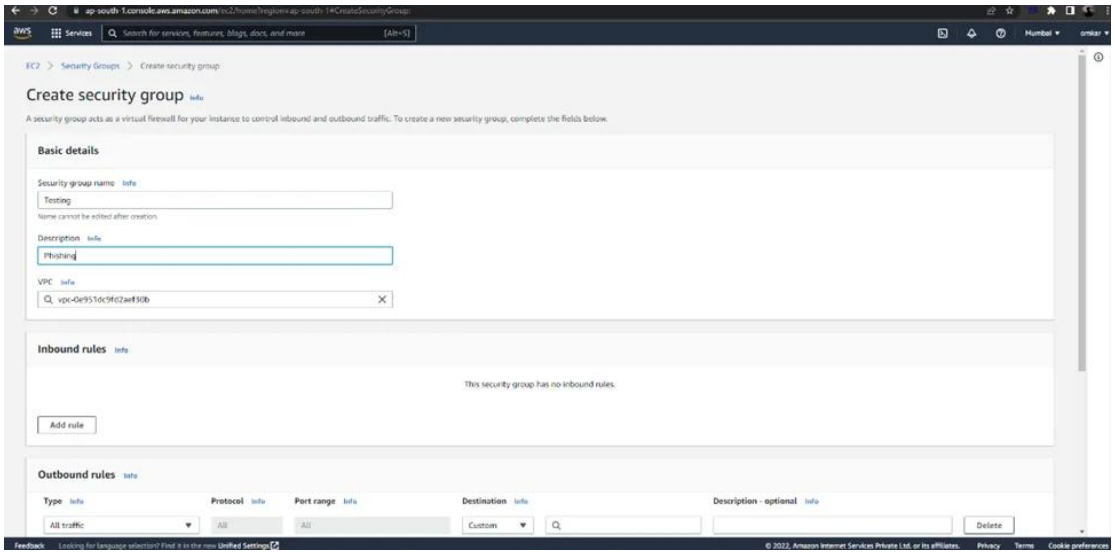
From EC2 Dashboard, select the Security Group under Network & Security tab.



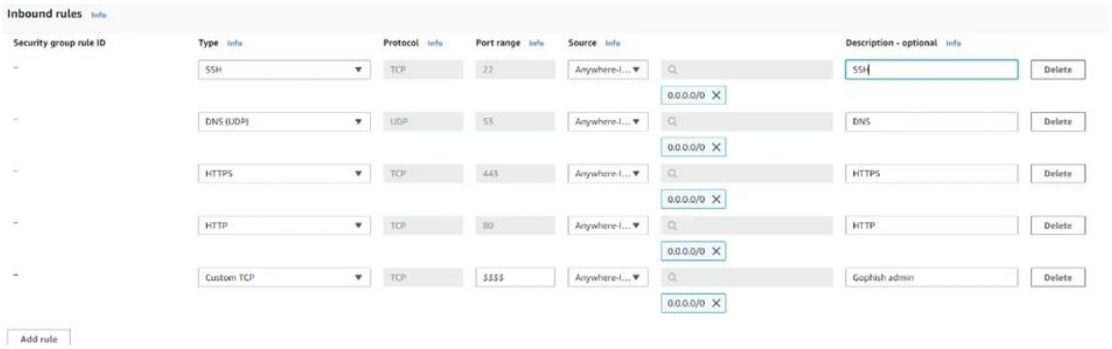
Now click on “Create security group”.



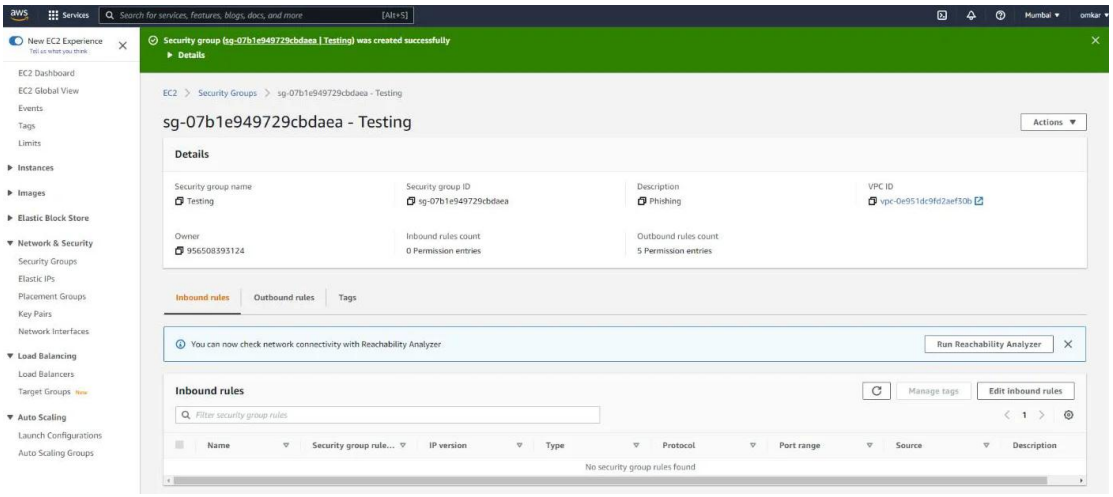
Enter the “Basic Details”.



As can be seen below, add the following Inbound Rules for SSH, DNS, HTTP, HTTPS, and gophish.

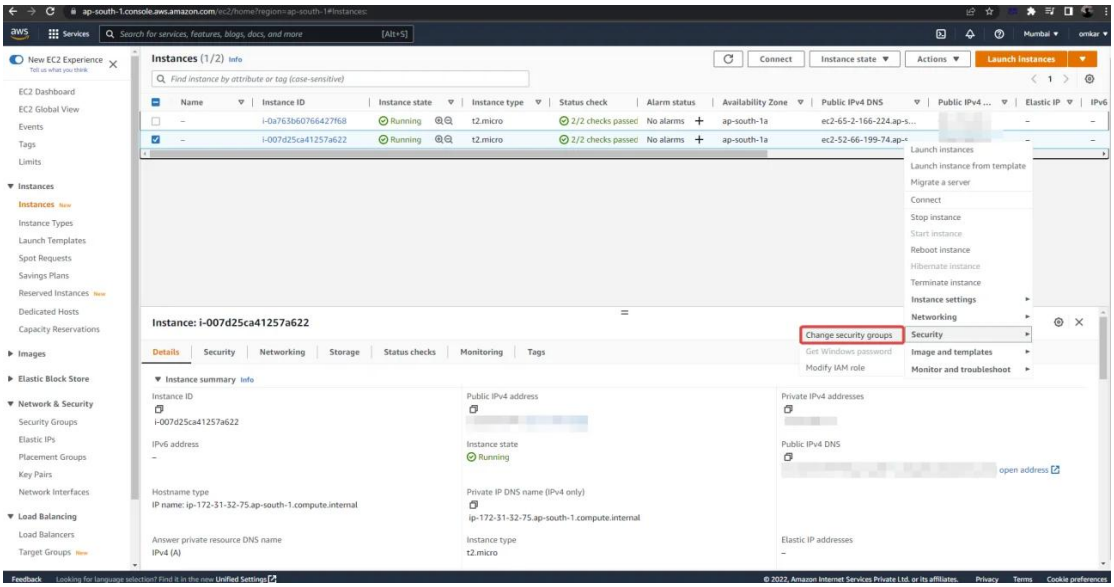


As can be seen below, a new Security Group (Testing) has been created successfully.

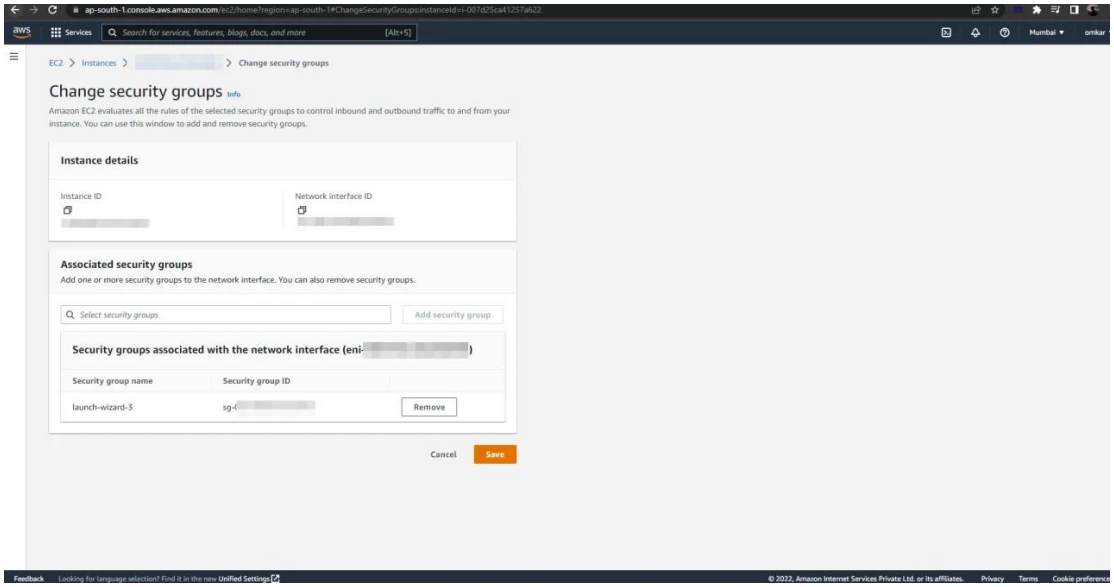


This newly created Security Group must now be assigned to the Ubuntu EC2 instance.

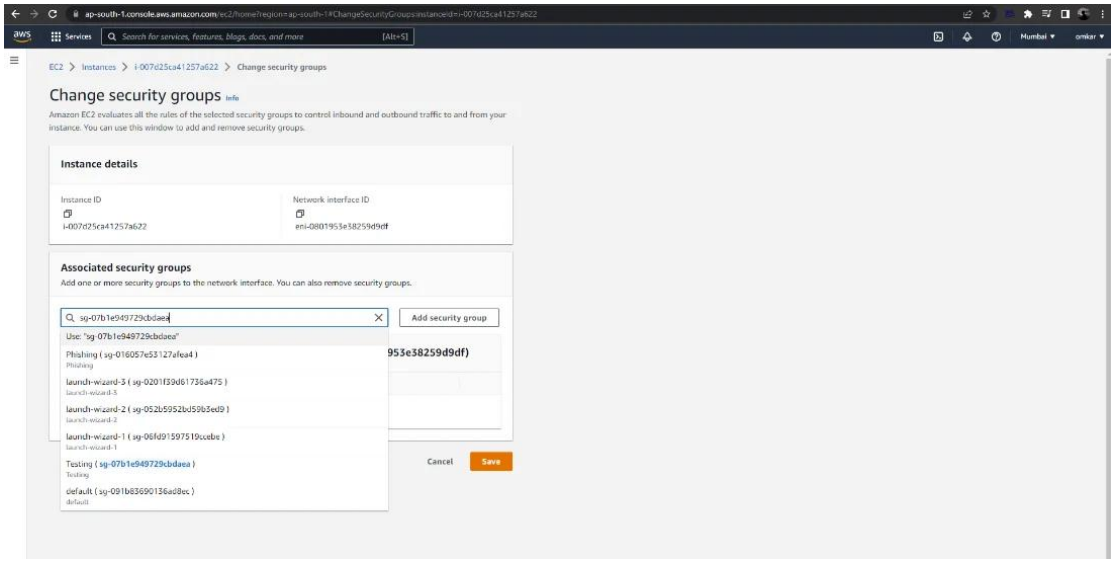
Next, Click on Navigate to Instances -> Right-click on an EC2 Instance -> Security –> Change security groups



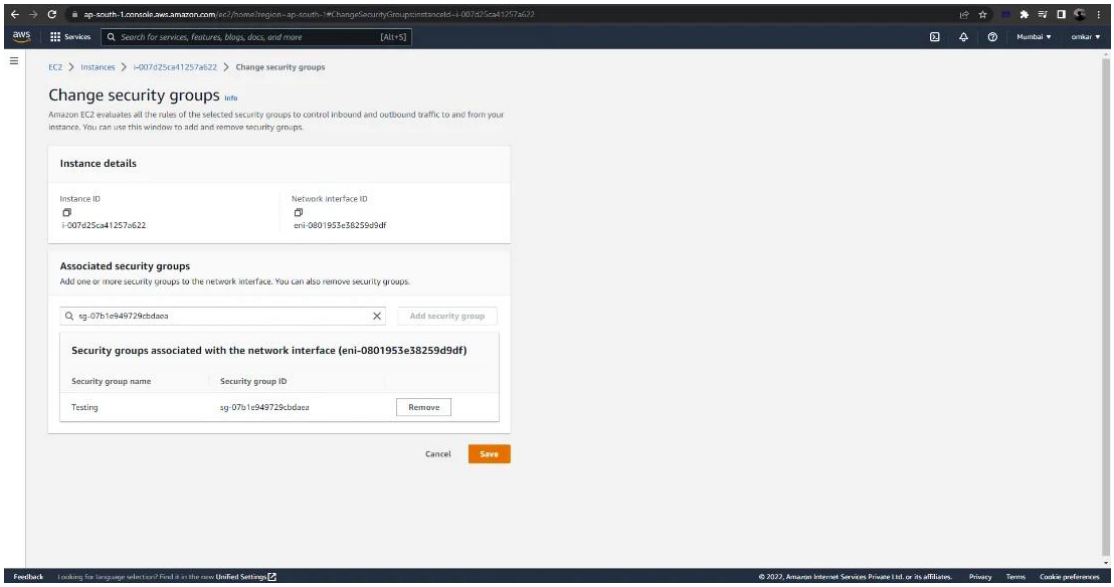
Remove the default Security Group (launch-wizard-3).



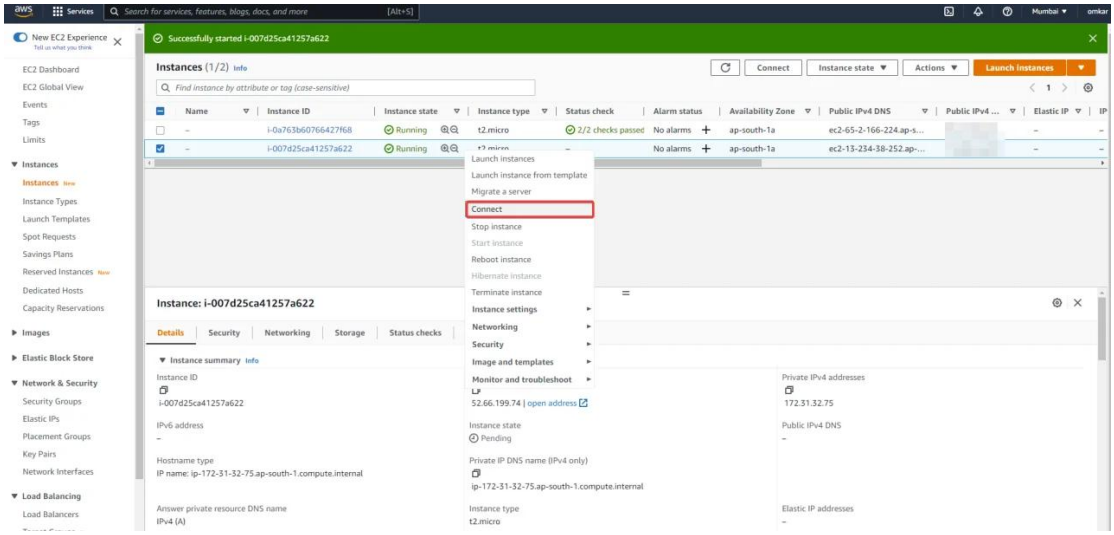
Select the newly created security group from the drop-down menu.



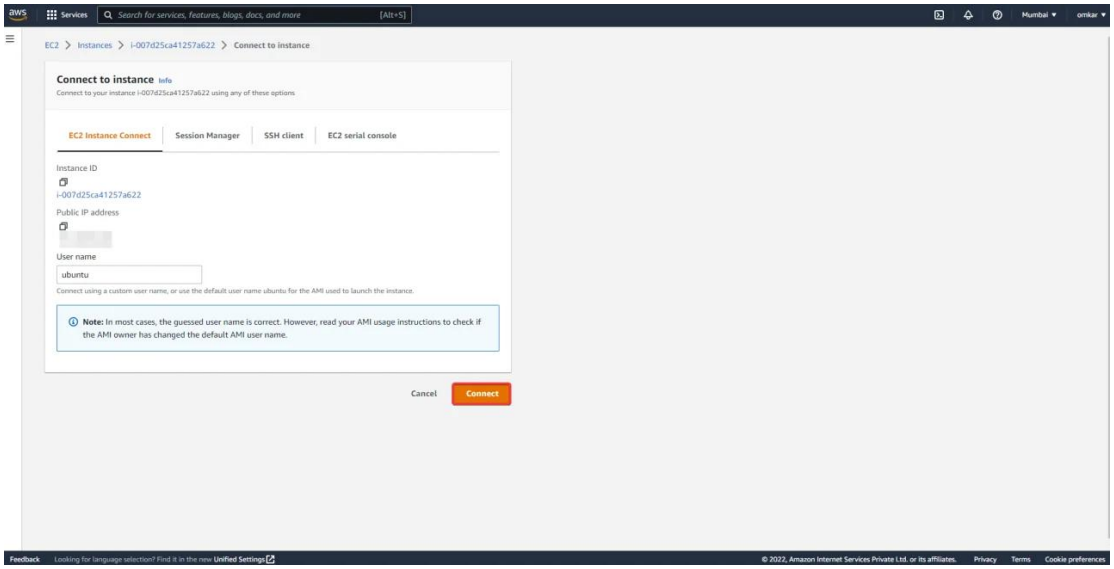
To save the changes, click on the “Save” button.

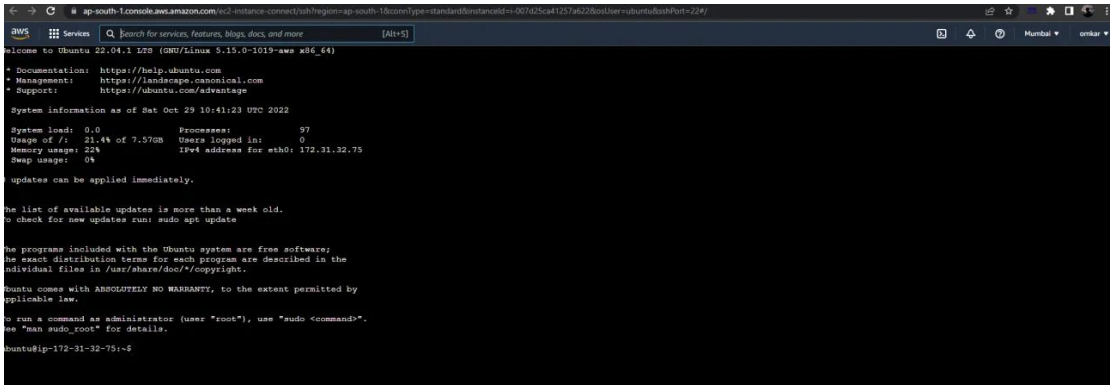


Next, right-click on an EC2 instance and select “Connect”.



Now click on “Connect”. This will redirect you to a web terminal session on our EC2 instance.

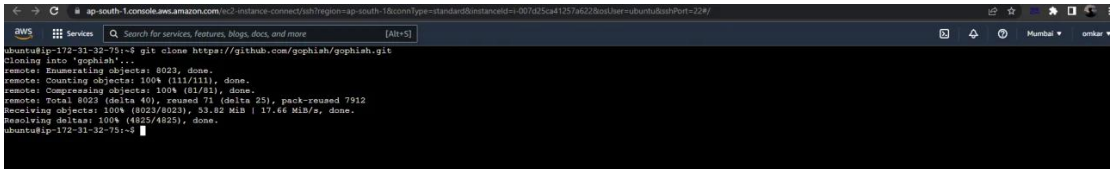




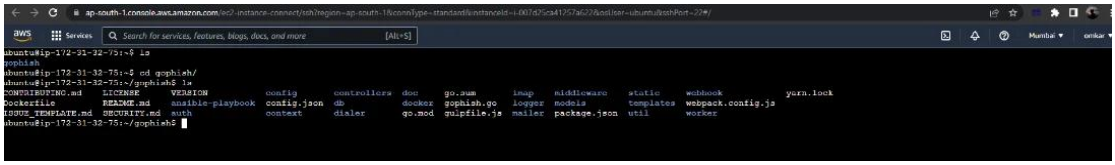
Installing gophish

gophish is a free and open-source phishing toolkit for enterprises and penetration testers. It enables the rapid and easy setup and execution of phishing interactions and security awareness training. Simply run to build gophish from source.

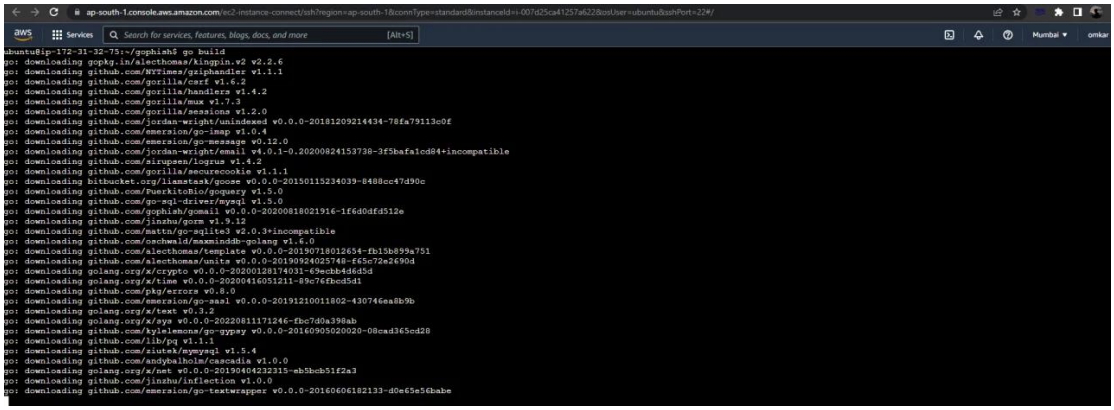
```
git clone <https://github.com/gophish/gophish.git>
```



Navigate to the project's source directory.

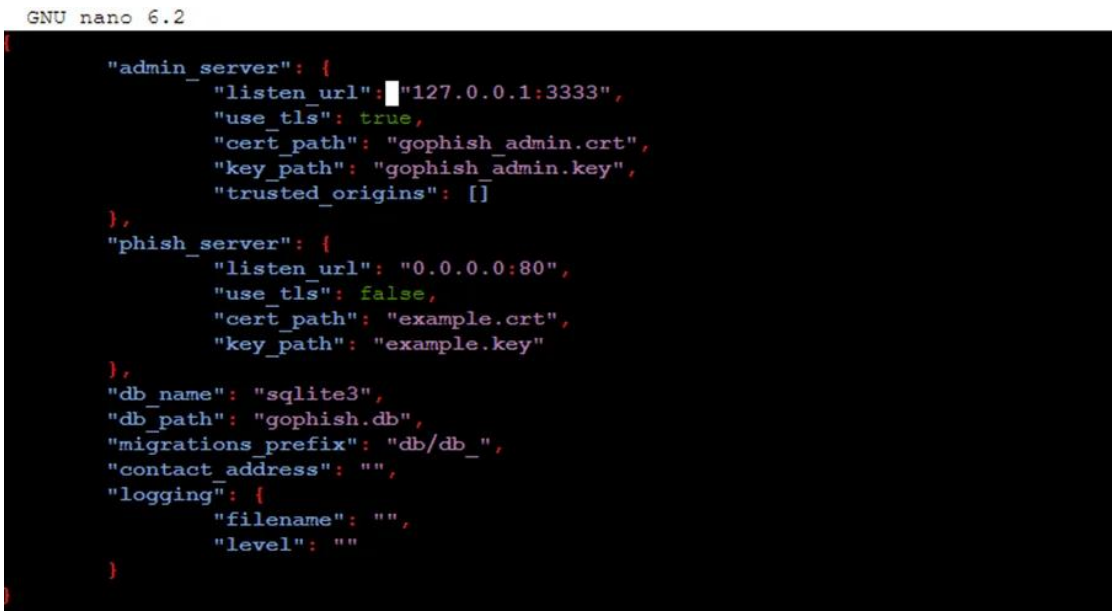


Then, execute go build.



Following that, you should have a binary named gophish in your current directory.

Replace 127.0.0.1:3333 with 0.0.0.0:3333 in the configuration file.



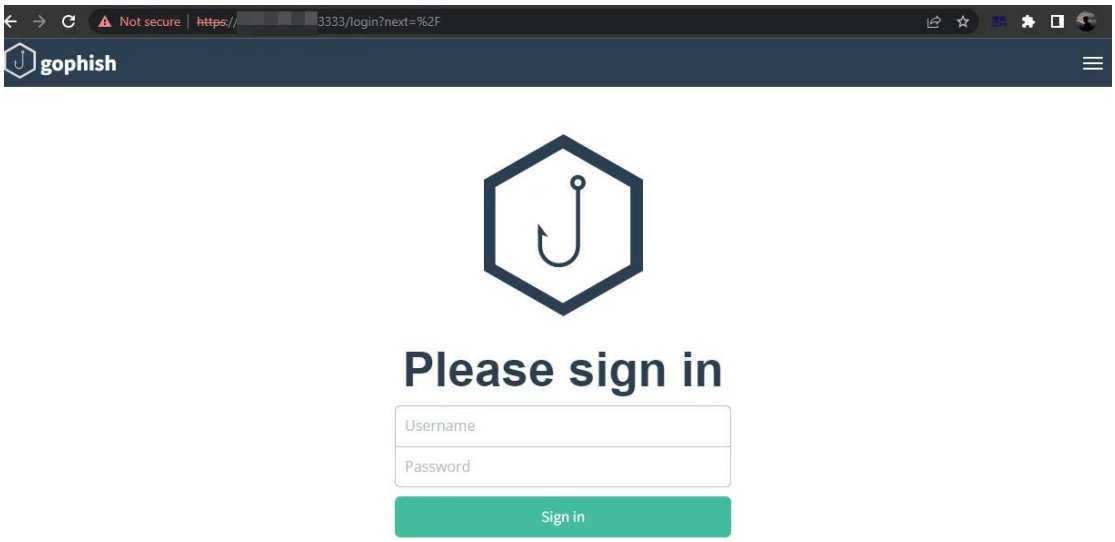
```
ubuntu@ip-172-31-32-75:~/gophish$ sudo nano config.json
ubuntu@ip-172-31-32-75:~/gophish$ sudo cat config.json
{
        "admin_server": {
                "listen_url": "0.0.0.0:3333",
                "use_tls": true,
                "cert_path": "gophish_admin.crt",
                "key_path": "gophish_admin.key",
                "trusted_origins": []
        },
        "phish_server": {
                "listen_url": "0.0.0.0:80",
                "use_tls": false,
                "cert_path": "example.crt",
                "key_path": "example.key"
        },
        "db_name": "sqlite3",
        "db_path": "gophish.db",
        "migrations_prefix": "db/db_",
        "contact_address": "",
        "logging": {
                "filename": "",
                "level": ""
        }
}
ubuntu@ip-172-31-32-75:~/gophish$
```

Execute the gophish file by using this command

```
sudo ./gophish
```

```
aws  Services  Search  [Alt+S]
ubuntu@ip-172-31-32-75:~/gophish$ sudo ./gophish
time="2022-11-03T06:55:44Z" level=warning msg="No contact address has been configured."
time="2022-11-03T06:55:44Z" level=warning msg="Please consider adding a contact_address entry in your config.json"
goose: no migrations to run. current version: 20220321133237
time="2022-11-03T06:55:44Z" level=info msg="Please login with the username admin and the password cf7056c92ef93367"
time="2022-11-03T06:55:44Z" level=info msg="Starting admin server at https://0.0.0.0:3333"
time="2022-11-03T06:55:44Z" level=info msg="Background Worker Started Successfully - Waiting for Campaigns"
time="2022-11-03T06:55:44Z" level=info msg="Starting IMAP monitor manager"
time="2022-11-03T06:55:44Z" level=info msg="Starting new IMAP monitor for user admin"
time="2022-11-03T06:55:44Z" level=info msg="Starting phishing server at http://0.0.0.0:80"
```

Now enter the ec2 instant IP address with the port number 3333. To access the gophish dashboard, go to https://ipaddress:3333 (be sure to include the https://).



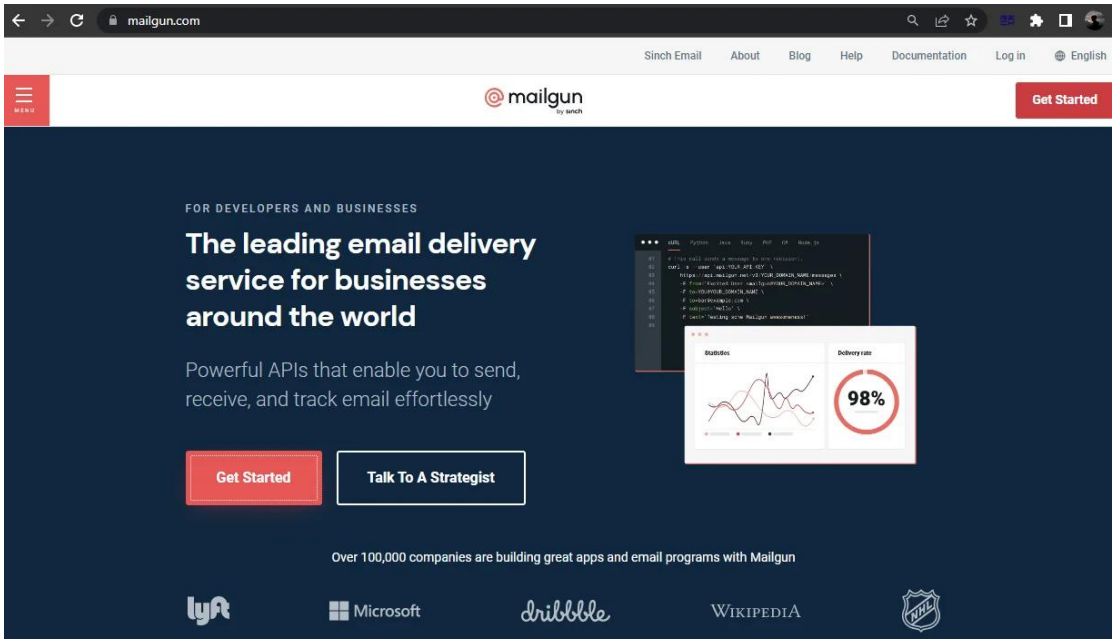
The password will be displayed in the terminal.

Let us now login to the gophish server.

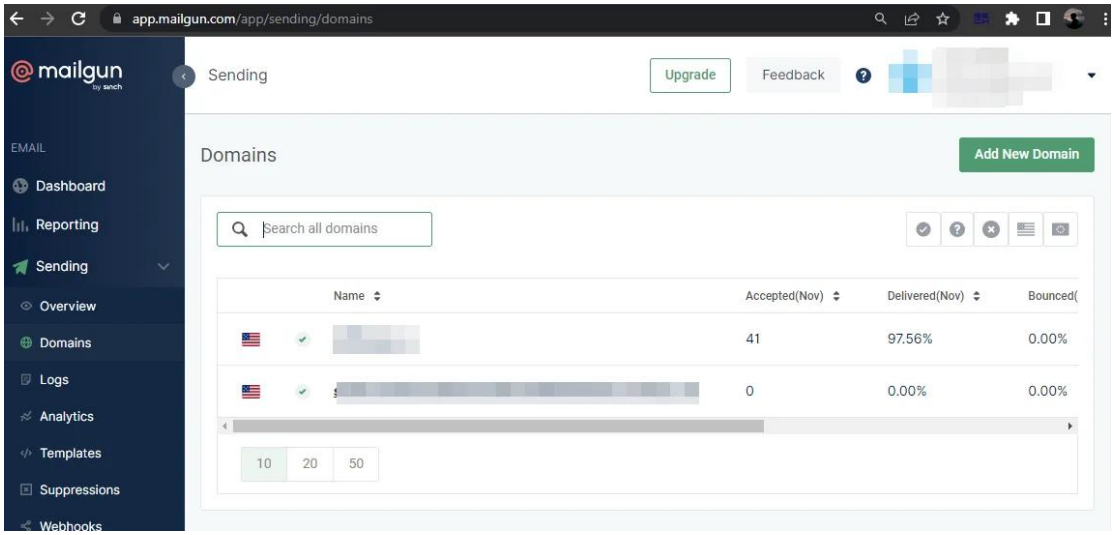
Setting up Mail Server

Create an account with MailGun.

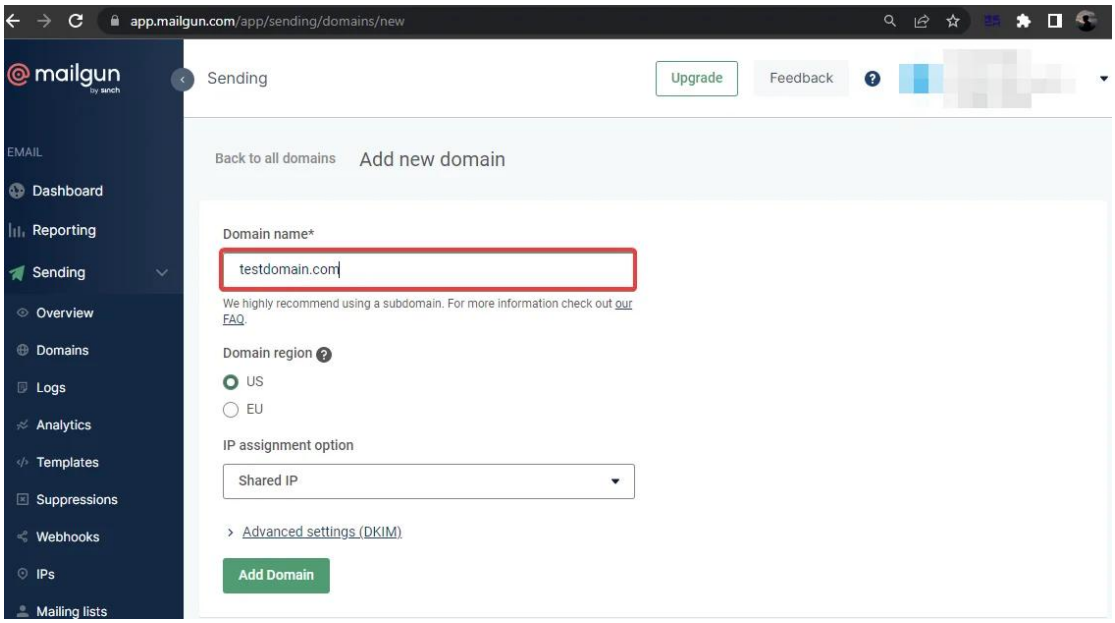
Note: Only If the credit card verification is successful, Mailgun will allow you to add domains.



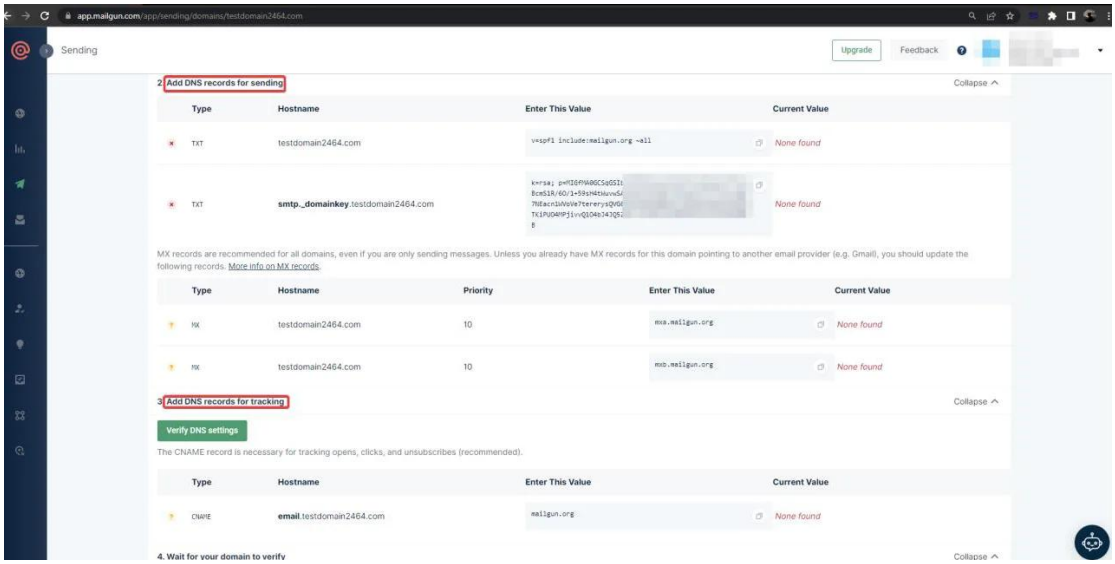
Navigate to Sending -> Domains, as seen below.



Add new domain in the “Domain name” field.

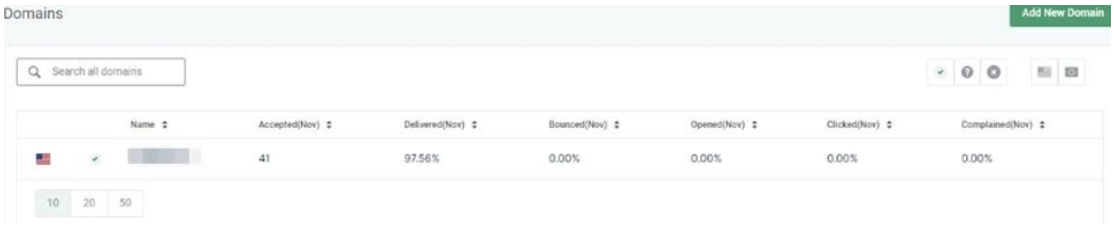


For sending emails, we must add the DNS record.



Verify the server after adding this DNS record to the Domain DNS server.

You will see a green tick after adding the DNS record in the domain config.



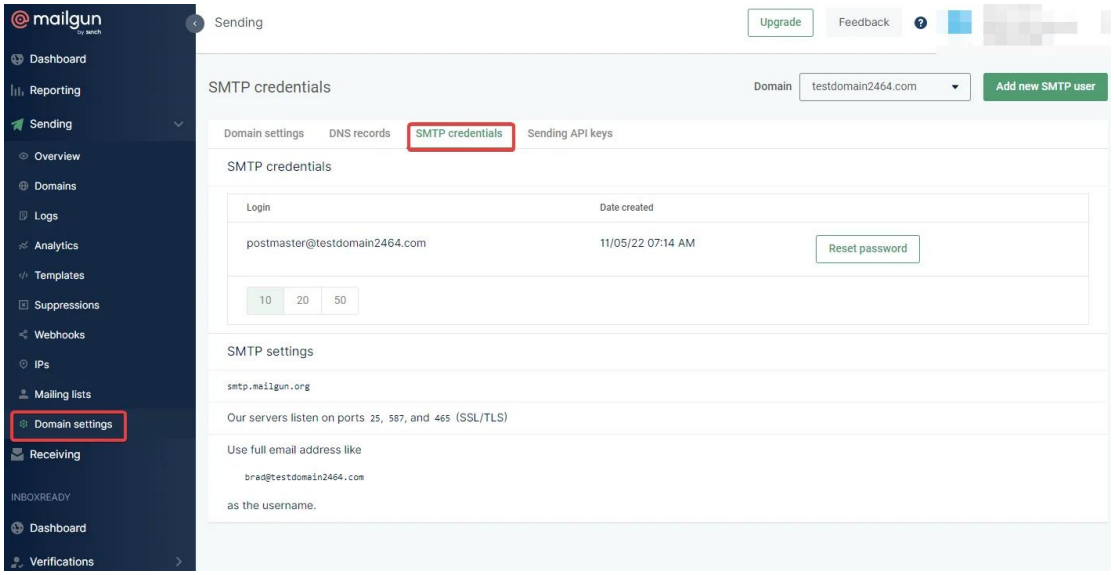
Let us now configure gophish using Mailgun.

Integrating Mailgun with gophish

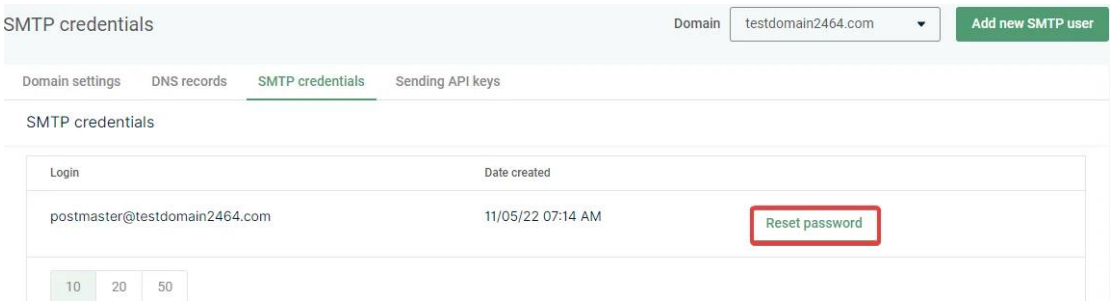
Creating SMTP Credentials

Mailgun requires DNS verification for the domain.

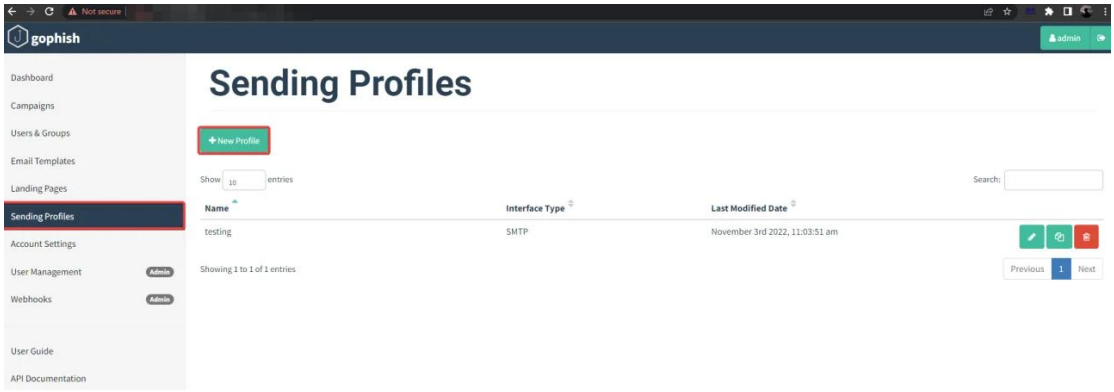
Select SMTP from the domain settings.



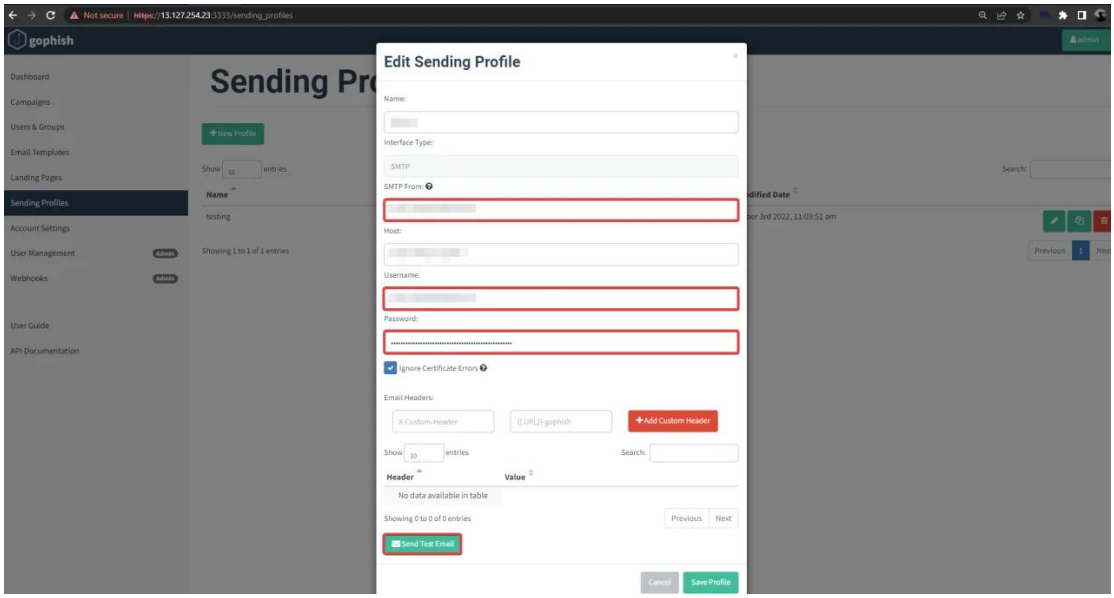
Click on Reset Password and then the password will get copied to the clipboard.



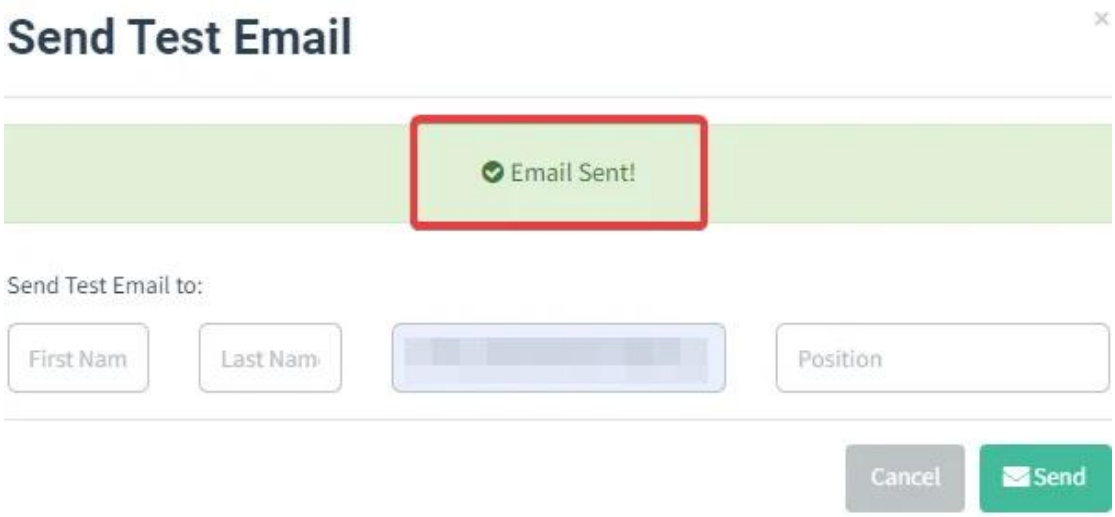
Now go to the gophish application and navigate to “Sending Profiles”.



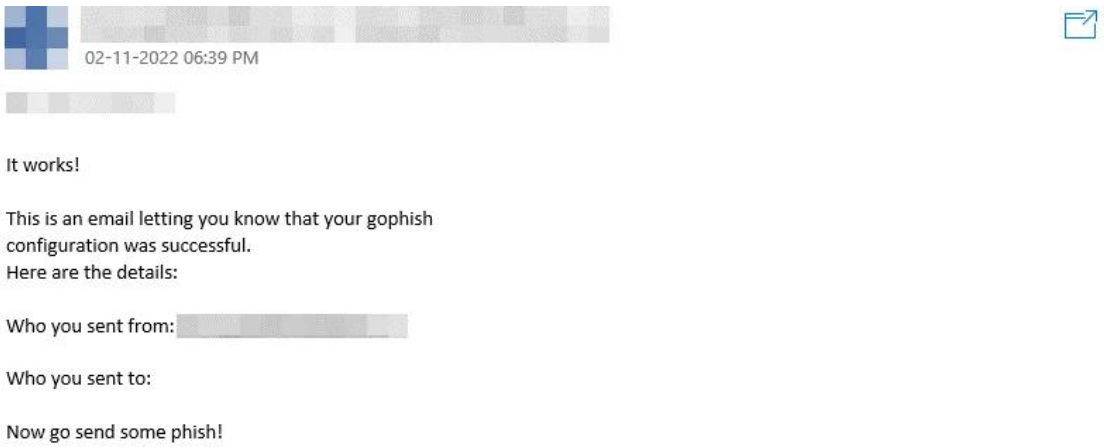
Now, select “New Profile”, enter your SMTP login and password, and send a test email.



The next step is to send a test email.

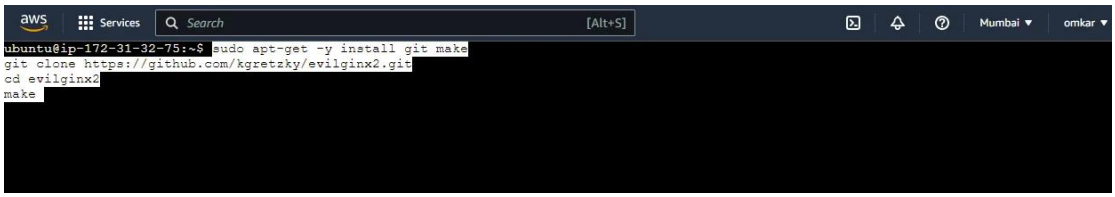


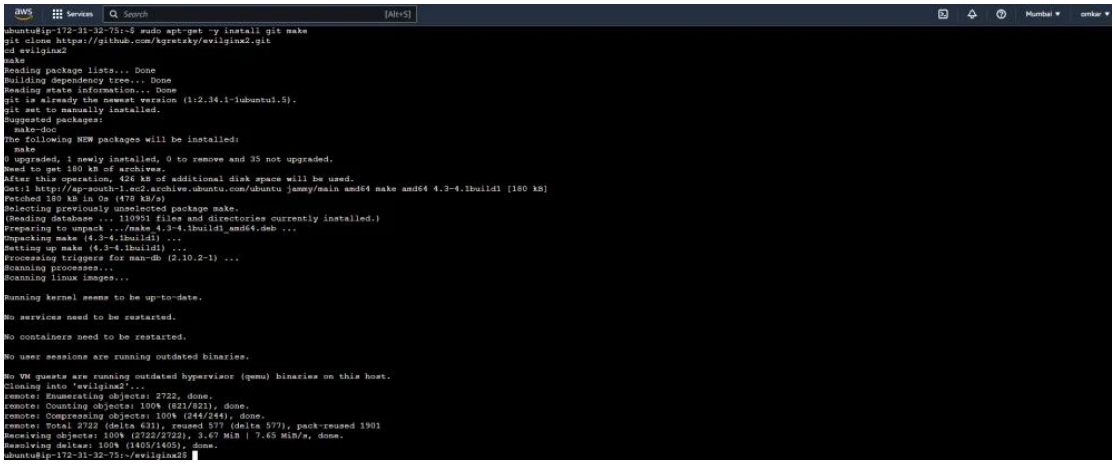
This is how the test mail will look like.



Setting up evileginx2

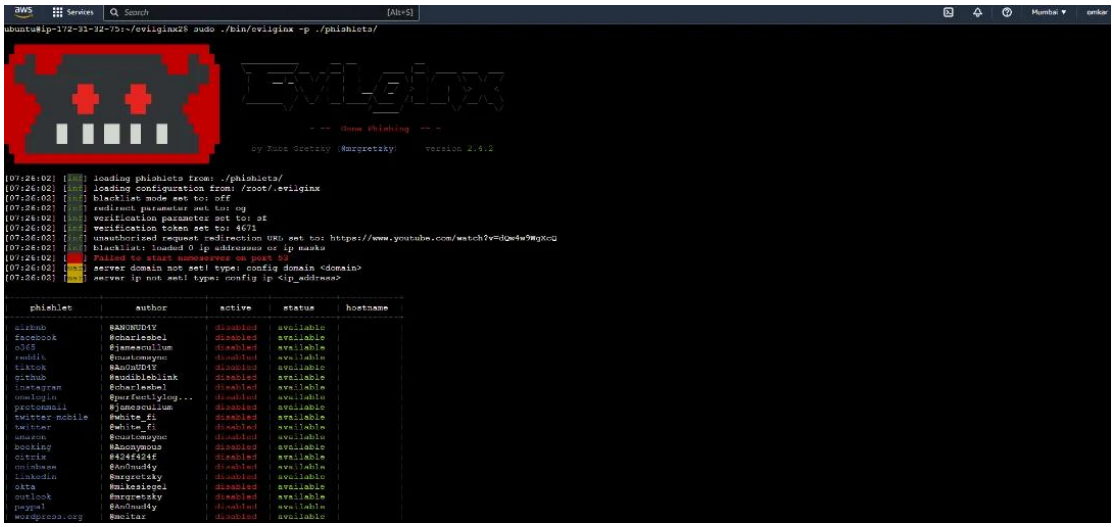
Install evilginx2 using the following command.





You can now launch evilginx2 from the local directory as follows:

```
sudo ./bin/evilginx -p ./phishlets/
```

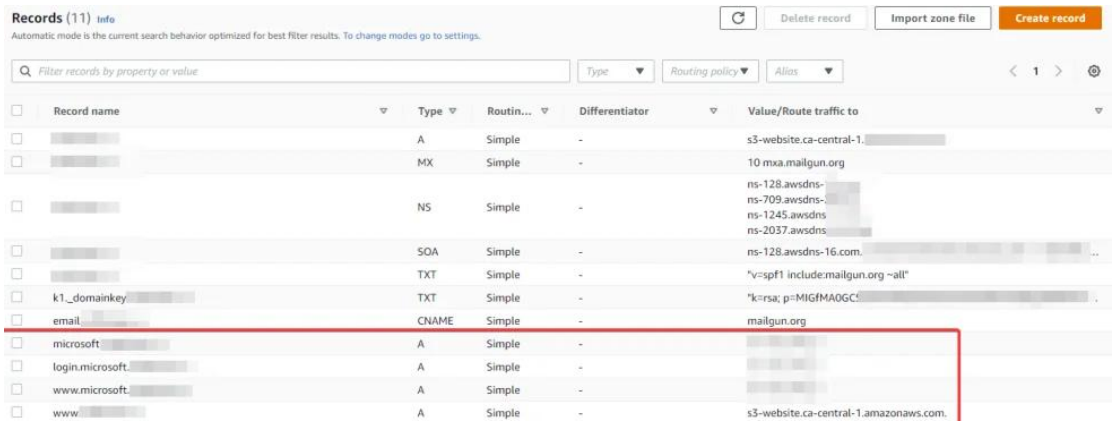


Configure the domain name and IP address.

config domain testdomain.com #the domain you bought for phishing

config ip <IP> #ec2 public Ip address

Navigate to Route53 and create a few **"A records"** as shown below.



Next, we need to configure the Office365 phishlet to match our domain:

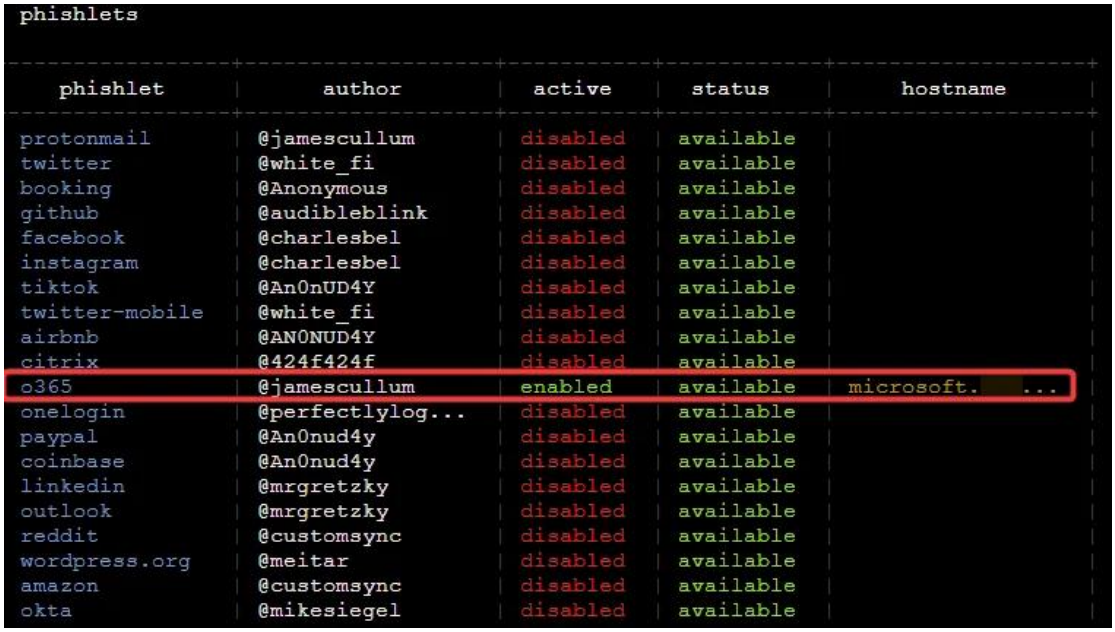
To add the phishlets.

```
phishlets hostname o365 microsoft.testdomain.com
```

To enable the o365 phishlets.

```
phishlets enable o365
```

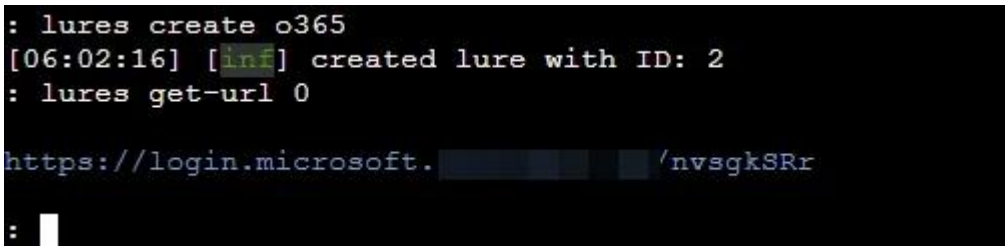
We can verify if the phishlet has been enabled by typing phishlets again:



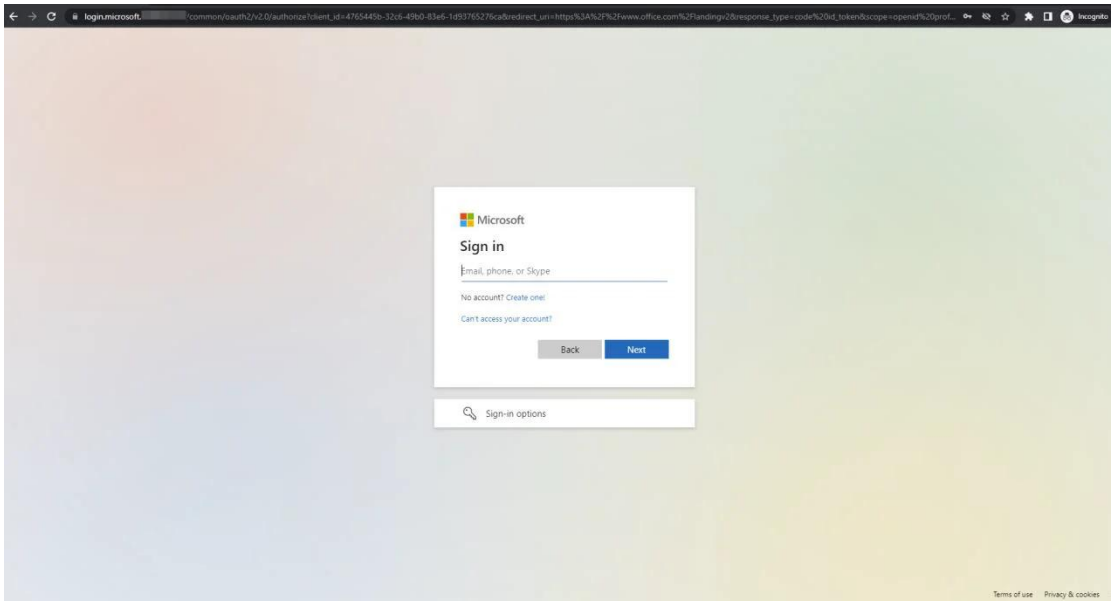
We now need a link that the victim clicks on, in evilginx2, the term for the link is "Lures".

Create lures and generate URLs using the following command.

```
lures create o365 # You will receive a lures ID after running this
command:
lures get-url 0
```



Our phishlet is now active and can be accessed by the URL https://login.microsoft.testdomain.com/nvsgkSRr.



When a victim inputs valid credentials, those credentials are recorded in the logs.

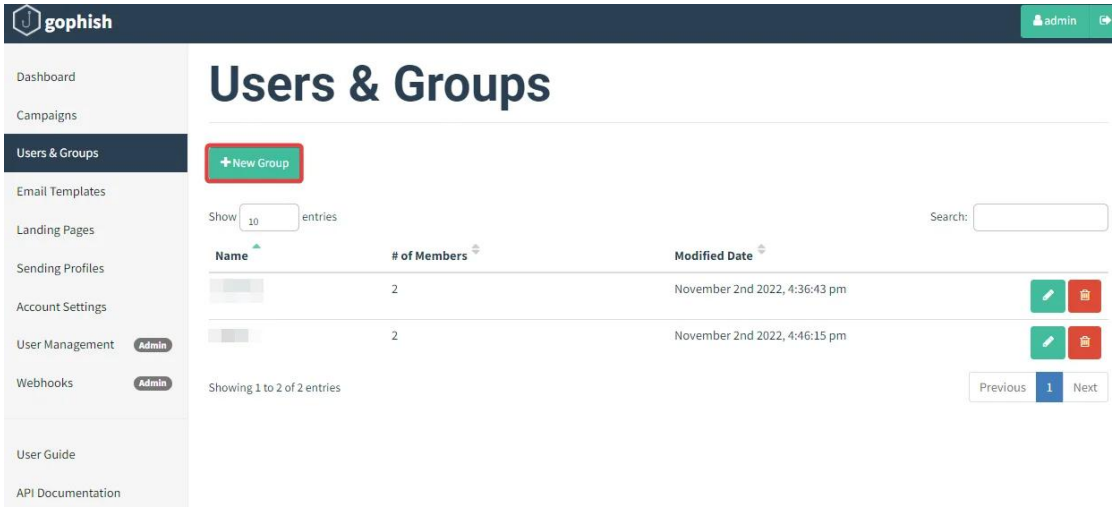
Next, type “sessions”. This lists all captured sessions.

```
sessions # list all captured sessions
```

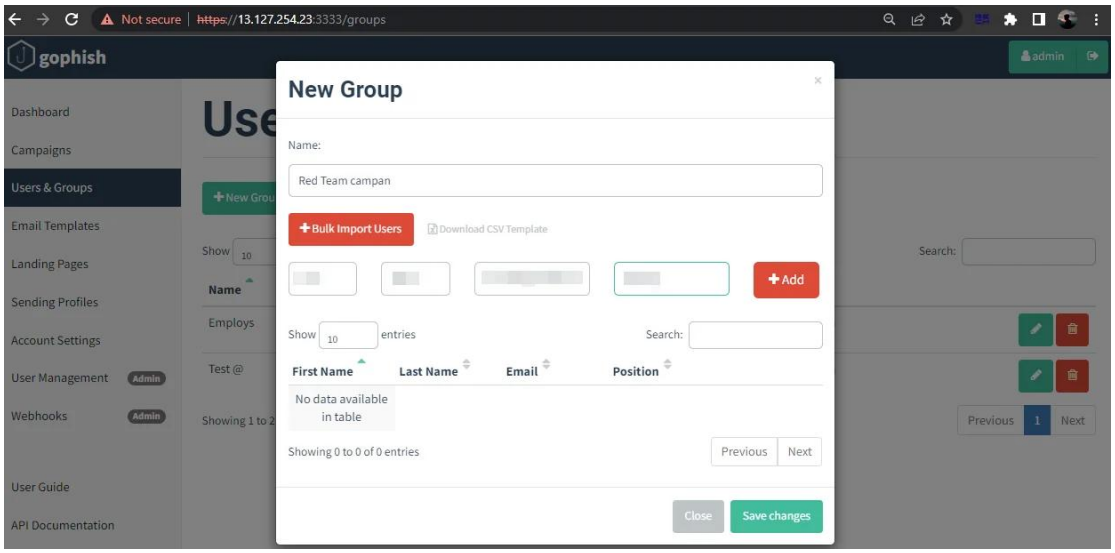
Running a gophish campaign

Log in to your gophish application.

Go to “Users & Groups” and click on “New Group”.

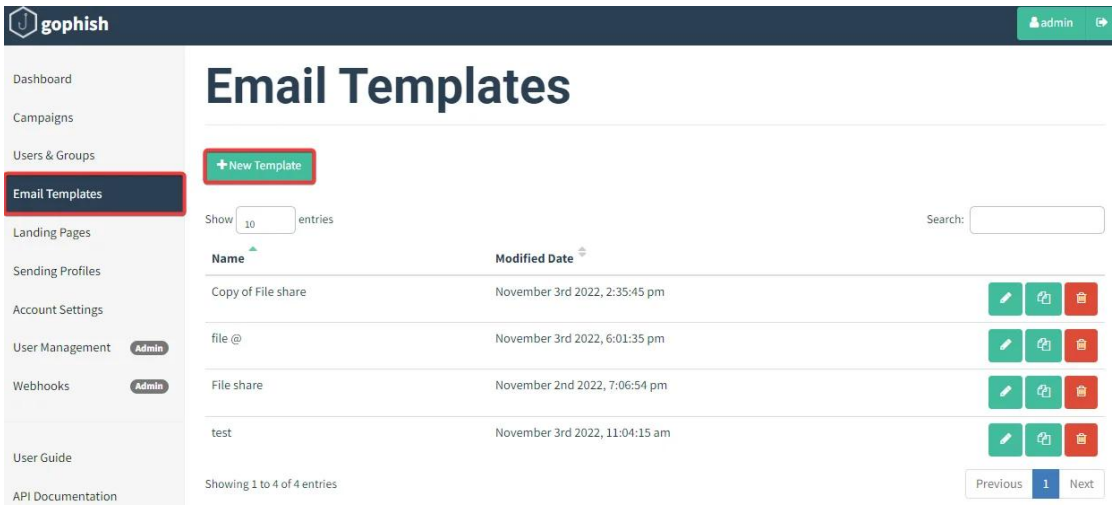


Enter the details of the targeted audience along with the email address.

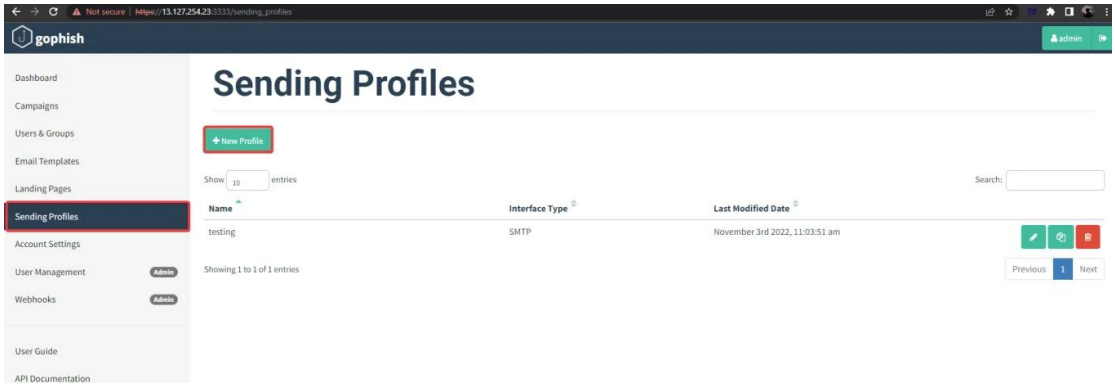


Next step is to create an email template.

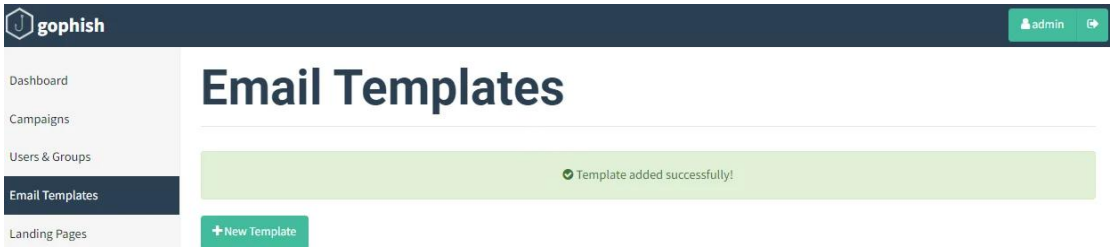
Click on Email Templates -> New Template.



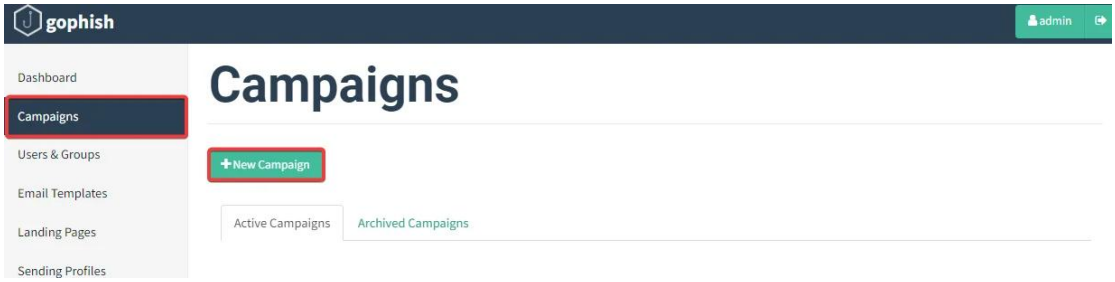
Give the template a suitable name and create a custom page as shown below.

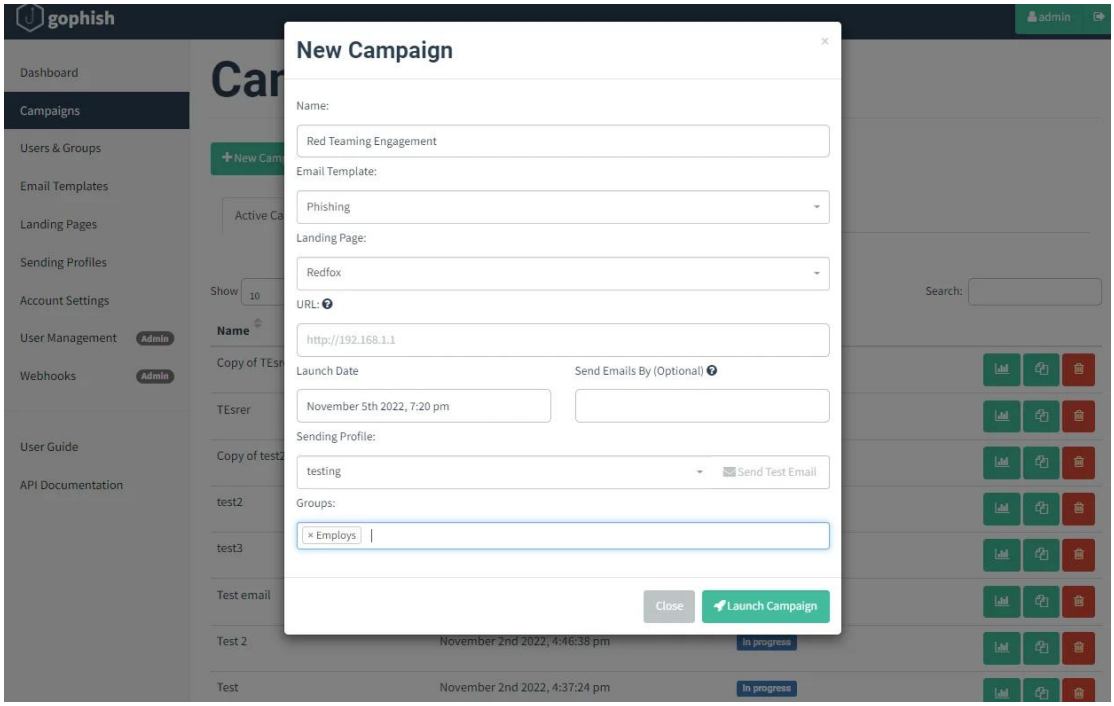


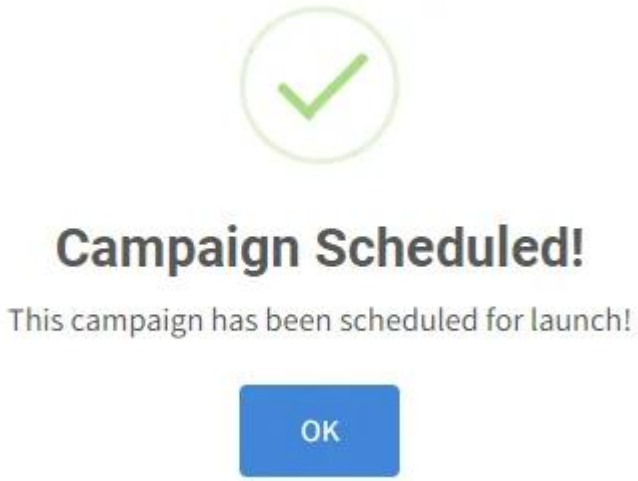
Click on “Save Template”.



Navigate to Campaigns -> New Campaign.







Once the target users submit their credentials, the evilginx2 sessions can be watched and used to replay the cookie sessions through the browser. It may take a few minutes for the victim to notice the malicious email in their inbox. When the phishing campaign begins, you should be transferred to a page that displays the campaign's outcomes. You have completed all of your tasks, and all you can do now is wait and hope that one of your emails gets opened by a victim. If the email was sent successfully, the victim should be able to receive it. Remember that spam filters and other email defences may attempt to block your emails or tell a system administrator that your email appears suspect. If this happens, experiment with less suspicious email templates, payloads, landing pages, and sending profiles.

Before delivering their best payloads, most penetration testers will try to investigate their target's phishing mitigations by sending simple payloads to see if they are blocked. For example, they may send a few test emails to see if macro-enabled Word or Excel files, malicious links, or custom binaries are banned. Whether particular

payloads are being prohibited by your target's email server, you can see if there is any way to get around these defences. If you want to adopt this method, you should create numerous email templates for each sort of test.